What is claimed is:

1. A method for balancing the load of a parallel processing system having a plurality of parallel processing elements arranged in a loop, wherein each processing element has a local number of tasks associated therewith, wherein r represents the number for a selected processing element PE_r, and wherein each of said processing elements are operable to communicate with a clockwise adjacent processing element and with an anti-clockwise adjacent processing element, the method comprising:

   determining within each of said processing elements a total number of tasks present within said loop;

   calculating a local mean number of tasks within each of said plurality of processing elements;

   calculating a local deviation within each of said plurality of processing elements;

   determining a sum weighted deviation within each of said processing elements for one-half of said loop in an anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

   determining a sum weighted deviation within each of said processing elements in one-half of said loop in a clockwise direction, said one-half of said loop being relative to each of said selected processing element;

   determining a clockwise transfer parameter and an anti-clockwise transfer parameter within each of said processing elements; and

   redistributing tasks among said plurality of processing elements in response to said clockwise transfer parameters and said anti-clockwise parameters within each of said plurality of processing elements.

2. The method of claim 1 wherein said determining within each of said processing elements a total number of tasks present within said loop, comprises:

   transmitting said local number of tasks associated with each of said processing elements to each other of said plurality of processing elements within said loop;

   receiving within each of said processing elements said number of local tasks associated with said each other of said plurality of processing elements; and

   summing said number of local tasks associated with each of said processing elements with said number of local tasks associated with each other of said plurality of processing elements.

3. The method of claim 1 wherein said determining said total number of tasks present within said loop includes solving the equation $V = \sum_{i=-N}^{i=N-1} v_i$, where V represents said total number of tasks, $2N$ represents the number of processing elements in said loop, and v_i represents said local number of tasks associated with an i^{th} processing element in said loop.

4. The method of claim 1 wherein said calculating a local mean number of tasks within each of said plurality of processing elements (PE_r) includes solving the equation $M_r = Trunc((V + E_r)/2N)$, where M_r is said local mean for PE_r, where $2N$ is the total number of processing elements in said loop, and where E_r is a number in the range of 0 to $(2N-1)$ and wherein each processing element has a different E_r value.

5. The method of claim 4 wherein E_r controls said *Trunc* function such that said total number of tasks (V) for said loop is equal to the sum of the local mean number of tasks (M_r) for each of said plurality of processing elements in said loop (i.e., $V = \sum_{i=-N}^{i=N-1} M_i$).

6. The method of claim 4 wherein said local mean $M_r = Trunc((V + E_r)/N)$ for each local PE_r within said loop is equal to one of X and $(X+1)$.

7. The method of claim 1 wherein said calculating a local deviation within each of said plurality of processing elements comprises finding the difference between said local number of tasks and said local mean number for each of said plurality of processing elements.

8. The method of claim 1 wherein said determining a sum weighted deviation within each of said processing elements for one-half of said loop in an anti-clockwise direction comprises:

assigning a weight to each other of said plurality of processing elements within said loop;

transmitting said local deviation and said weight associated with each of said processing elements half way around said loop in an anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

receiving said local deviation and said weight associated with each other of said processing elements half way around said loop in a clockwise direction, said one-half of said loop being relative to each of said selected processing elements; and

summing the product of said local deviation and said weight associated with each other of said processing elements half way around said loop in a clockwise direction.

9. The method of claim 1 wherein said determining a sum weighted deviation within each of said processing elements in one-half of said loop in a clockwise direction comprises:

assigning a weight to each other of said plurality of processing elements within said loop;

transmitting said local deviation and said weight associated with each of said processing elements half way around said loop in an clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

receiving said local deviation and said weight associated with each other of said processing elements half way around said loop in a anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements; and

summing the product of said local deviation and said weight associated with each other of said processing elements half way around said loop in a anti-clockwise direction.

10. The method of claim 1 wherein said determining a clockwise transfer parameter and an anti-clockwise transfer parameter within each of said processing elements comprises:

setting $T_a = (S/4) - \Delta$; and

setting $T_c = (S/4) + \Delta$, where T_c represents said clockwise transfer parameter, T_a represents said anti-clockwise transfer parameter, $\Delta = (A - C)/4N$, A represents the sum weighted deviation within each of said processing elements in one-half of said loop in an anti-clockwise direction, C represents sum weighted deviation within each of said processing elements in one-half of said loop in a clockwise direction, and N represents the number of PEs on the loop.

11. The method of claim 1 wherein said determining a clockwise transfer parameter and an anti-clockwise transfer parameter within each of said processing elements comprises at least one of:

setting $T_c = Trunc[(2S + \Delta) \div 4]$ and $T_a = S - T_c$; and

setting the $T_a = Trunc[(2S - \Delta) \div 4]$ and $T_c = S - T_a$;

where T_c represents said clockwise transfer parameter, where T_a represents said anti-clockwise transfer parameter, where Δ = Mag, if Δ > Mag, where Δ = –Mag, if Δ < -Mag, where Mag = abs($2S$), and where S represents the local deviation of a selected processing element.

12. A method for reassigning tasks among an odd numbered plurality of processing elements within a parallel processing system, said processing elements being connected in a loop and each having a local number of tasks associated therewith, the method comprising:

determining a total number of tasks on said loop;

computing a local mean value for a selected processing element;

computing a local deviation for said selected processing element, said local deviation representative of the difference between said local number of tasks for said selected processing element and said local mean value for said selected processing element;

inserting a phantom processing element within said loop;

assigning a weight to each of said plurality of processing elements;

summing a weighted deviation of said processing elements located within one-half of the loop in an anti-clockwise direction relative to said selected processing element;

summing said weighted deviation of said processing elements located within one-half of the loop in a clockwise direction relative to said selected processing element;

computing a number of tasks to transfer in a clockwise direction for said selected processing element;

computing a number of tasks to transfer in an anti-clockwise direction for said selected processing element; and

reassigning tasks relative to the said number of tasks to transfer in a clockwise direction and said number of task to transfer in an anti-clockwise direction.

13. The method of claim 12 wherein said determining the total number of tasks on said loop, comprises:

transmitting said local number of tasks associated with each of said processing elements to each other of said plurality of processing elements within said loop;

receiving within each of said processing elements said number of local tasks associated with said each other of said plurality of processing elements; and

summing said number of local tasks associated with each of said processing elements with said number of local tasks associated with each other of said plurality of processing elements.

14. The method of claim 12 wherein computing a local mean value for a selected processing element includes solving the equation $M_r = Trunc((V + E_r)/2N)$, where M_r is said local mean for a selected PE_r, $2N$ is the total number of processing elements in said loop, and E_r is a number in the range of 0 to $(2N-1)$ and wherein each processing element has a different E_r value.

15. The method of claim 14 wherein E_r controls said *Trunc* function such that said total number of tasks (V) for said loop is equal to the sum of the local mean number of tasks (M_r) for each of said plurality of processing elements in said loop (i.e., $V = \sum_{i=0}^{i=2N-1} M_i$).

16. The method of claim 12 wherein said inserting a phantom processing element within said loop further comprises:

locating said phantom processing element in a position within said loop that is diametrically opposed to said processing element; and

assigning a zero deviation value to said phantom processing element.

17. The method of claim 12 wherein said assigning a weight to each of said plurality of processing elements includes assigning a weight dependent upon each of said processing element's location to said selected processing element.

18. The method of claim 12 wherein said computing a local mean value for a selected processing element, said computing a local deviation for said selected processing element, said inserting a phantom processing element within said loop, said assigning a weight to each of said plurality of processing elements, said summing said weighted deviation of said processing elements located within one-half of the loop in an anti-clockwise direction, summing said weighted deviation of said processing elements located within one-half of the loop in a clockwise direction, computing a number of tasks to transfer in a clockwise direction for said selected processing element, computing a number of tasks to transfer in an anti-clockwise direction for said selected processing element, and reassigning tasks relative to the said number of tasks

to transfer in a clockwise direction and said number of tasks to transfer in an anti-clockwise direction are completed simultaneously for each of said plurality of processing elements within said loop.

19. The method of claim 12 wherein said summing said weighted deviation of said processing elements located within one-half of the loop in an anti-clockwise direction relative to said selected processing element comprises·

transmitting said local weighted deviation associated with each of said processing elements half way around said loop in an anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

receiving said local weighted deviation associated with each other of said processing elements half way around said loop in a clockwise direction, said one-half of said loop being relative to each of said selected processing elements; and

summing said local weighted deviations associated with each other of said processing elements half way around said loop in a clockwise direction.

20. The method of claim 12 wherein summing said weighted deviation of said processing elements located within one-half of the loop in a clockwise direction relative to said selected processing element comprises:

transmitting said local weighted deviation associated with each of said processing elements half way around said loop in an clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

receiving said local weighted deviation associated with each other of said processing elements half way around said loop in a anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements; and

summing said local weighted deviations associated with each other of said processing elements half way around said loop in an anti-clockwise direction.

21. A memory device carrying a set of instructions which, when executed, perform a method comprising:

determining within each of said processing elements a total number of tasks present within said loop;

calculating a local mean number of tasks within each of said plurality of processing elements;

calculating a local deviation within each of said plurality of processing elements;

determining a sum weighted deviation within each of said processing elements for one-half of said loop in an anti-clockwise direction, said one-half of said loop being relative to each of said selected processing elements;

determining a sum weighted deviation within each of said processing elements in one-half of said loop in a clockwise direction, said one-half of said loop being relative to each of said selected processing element;

determining a clockwise transfer parameter and an anti-clockwise transfer parameter within each of said processing elements; and

redistributing tasks among said plurality of processing elements in response to said clockwise transfer parameters and said anti-clockwise parameters within each of said plurality of processing elements.